

Intégration d'informations spatiales floues dans le filtre particulaire : application au suivi de formes

Nicolas Widynski^{1,2}Séverine Dubuisson²Isabelle Bloch¹¹ Télécom ParisTech, CNRS UMR 5141 LTCI, Paris, France² UPMC - LIP6, Paris, France

Nicolas.Widynski@enst.fr, Severine.Dubuisson@lip6.fr, Isabelle.Bloch@enst.fr

Résumé

Dans le cadre du suivi d'objets dans les séquences vidéo, nous proposons d'estimer conjointement les paramètres (dynamiques, d'apparence), et l'état (la position), par une méthode séquentielle de Monte-Carlo, le filtre particulaire. Nous modélisons ensuite des informations spatiales caractérisées par un petit nombre de classes par des ensembles flous, qui peuvent être fixés ou appris. Par exemple, une information directionnelle est modélisée par quatre ensembles flous représentant les quatre directions cardinales (les classes dans ce cas), et un paramètre d'orientation (un angle) pourra appartenir à l'un de ces quatre ensembles flous. L'estimation de cette appartenance est alors prise en compte dans le filtre et l'algorithme de suivi. Cette méthodologie est générique dans le sens où elle permet de manipuler divers types d'informations floues (relations spatiales, contraintes de forme...) qui sont ensuite directement intégrées au filtre particulaire.

Mots Clef

Filtre particulaire, informations spatiales floues, suivi de formes floues.

Abstract

With the aim of object tracking in video sequences, we suggest to estimate both the parameters (dynamics, shape) and the state (position), by a sequential Monte-Carlo method, the particle filter. We then model spatial information characterized by a small number of classes by fuzzy sets, which can be fixed or learned. For instance, an information of direction is modeled by four fuzzy sets representing the four cardinal directions (the classes in this case), and a direction parameter (an angle) can belong to one of these fuzzy sets. The estimation of this membership is used in the filter and the tracking algorithm. This methodology is generic as it allows manipulating several kinds of fuzzy information (spatial relations, shape constraints) which are then directly integrated into the particle filter.

Keywords

Particle filter, fuzzy spatial relations, fuzzy shape tracking.

1 Introduction

Les méthodes séquentielles de Monte-Carlo ont connu un intérêt grandissant au milieu des années 1990, au moment où l'essor des nouvelles technologies rendait possible leur utilisation en pratique. Depuis, le champ d'applications n'a cessé de s'élargir : finance, ingénierie, biologie, physique, traitement du signal et de l'image... C'est dans ce dernier domaine que nous plaçons le contexte de notre travail. Dans cet article, nous nous intéresserons particulièrement au suivi de formes avec les méthodes séquentielles de Monte-Carlo, aussi appelées filtres particulaires. Les pionniers [5] proposèrent d'intégrer au filtre particulaire une forme moyenne apprise. Le suivi se fait ensuite en perturbant la forme de base par des transformations affines. Plus récemment, dans [2], les auteurs proposent de guider le suivi par filtre particulaire en utilisant des modèles d'apparence issus d'une base d'apprentissage et déformés par des opérations affines. Ces approches présentent le principal défaut d'utiliser un modèle de forme dynamique pauvre, en l'occurrence affine, ne permettant pas de gérer les déformations locales de la forme. Ainsi, dans [9] les auteurs intègrent des formes apprises représentées par ensembles de niveaux, et proposent des nouvelles formes pendant le suivi en effectuant une descente de gradient sur l'énergie associée à la forme de l'instant précédent. Toutefois cette approche est lourde en temps de calcul dans la mesure où chaque particule est liée à un ensemble de niveau. De plus la taille de la base d'apprentissage doit être conséquente afin de pouvoir capter toutes les variations de formes.

Notre approche consiste à définir l'appartenance des paramètres (par exemple dynamiques, de formes) à des ensembles flous (définissant par exemple des classes d'orientations, de postures). Ces ensembles flous peuvent être fixés ou appris, et sont générés selon des stratégies de fusion prédéfinies. De cette représentation générique, on est capable d'extraire automatiquement une dynamique entre ensembles flous, et d'illustrer des caractéristiques différentes sans avoir recours à une modélisation mathématique contraignante des paramètres. Nous proposons ainsi dans cet article d'intégrer des informations floues dans le filtre particulaire, ainsi qu'une application intégrant des relations

spatiales floues et des formes floues.

2 Méthodes séquentielles de Monte-Carlo

2.1 Filtrage bayésien

Considérons le problème classique du filtrage bayésien et notons $\mathbf{x}_t \in \mathcal{X}$ la variable d'état cachée au temps t d'un processus stochastique markovien et $\mathbf{y}_t \in \mathcal{Y}$ la variable observée. Le système se décrit par l'évolution temporelle de \mathbf{x}_t et par l'équation de mesure :

$$\mathbf{x}_t = g_t(\mathbf{x}_{t-1}, \mathbf{v}_t) \quad ; \quad \mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{w}_t) \quad (1)$$

où \mathbf{v}_t et \mathbf{w}_t sont des bruits blancs indépendants. Le suivi non linéaire bayésien consiste à estimer la distribution *a posteriori* $\Pr(d\mathbf{x}_t | \mathbf{y}_{1:t})$ selon les fonctions non linéaires g_t et h_t , respectivement de transition et de mesure.

2.2 Modélisation des paramètres

Il peut être intéressant d'intégrer les paramètres à l'estimation de la loi de filtrage. Une manière classique de faire cela consiste à décomposer le vecteur de paramètres $\mathbf{r}_t \in \mathcal{R}$ en R composantes indépendantes $\mathbf{r}_t = (\mathbf{r}_t^1, \dots, \mathbf{r}_t^R)$, et de considérer une dynamique de marche aléatoire pour chacune d'entre elles. Cela nous mène au système suivant :

$$\begin{aligned} \mathbf{x}_t &= g_t(\mathbf{x}_{t-1}, \mathbf{r}_t, \mathbf{v}_t) \\ \mathbf{r}_t &= \mathbf{r}_{t-1} + \mathbf{s}_t \\ \mathbf{y}_t &= h_t(\mathbf{x}_t, \mathbf{w}_t) \end{aligned} \quad (2)$$

avec $\mathbf{s}_t = (\mathbf{s}_t^r)_{r=1, \dots, R}$ un vecteur de bruits gaussiens mutuellement indépendants. En pratique, les hypothèses d'indépendance et de marche aléatoire sont classiques, et permettent de facilement modéliser la plupart des problèmes de suivi.

2.3 Filtre particulaire

Le filtrage bayésien avec la modélisation explicite des paramètres consiste en l'estimation de la loi *a posteriori* $\Pr(d\mathbf{x}_t, d\mathbf{r}_t | \mathbf{y}_{1:t})$. Sous les hypothèses markoviennes et d'indépendance conditionnelle des observations sachant le processus d'état, la densité de filtrage qui en résulte se compose d'une première étape de prédiction :

$$p(\mathbf{x}_t, \mathbf{r}_t | \mathbf{y}_{1:t-1}) = \int_{\mathcal{X}} \int_{\mathcal{R}} p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{r}_t) p(\mathbf{r}_t | \mathbf{r}_{t-1}) \times p(\mathbf{x}_{t-1}, \mathbf{r}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} d\mathbf{r}_{t-1} \quad (3)$$

puis d'une étape de filtrage :

$$p(\mathbf{x}_t, \mathbf{r}_t | \mathbf{y}_{1:t}) \propto p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{r}_t) p(\mathbf{x}_t, \mathbf{r}_t | \mathbf{y}_{1:t-1}) \quad (4)$$

Une forme analytique de la distribution *a posteriori* est disponible lorsque la fonction de transition de l'état et l'équation de mesure sont linéaires et que les bruits correspondants sont gaussiens, ce qui mène au filtre

de Kalman [6]. Dans le cas contraire, les filtres particuliers, ou méthodes séquentielles de Monte-Carlo, sont utilisés pour approcher la distribution *a posteriori* par une somme pondérée de N masses de Dirac $\delta_{\mathbf{x}_t^{(n)}, \mathbf{r}_t^{(n)}}(d\mathbf{x}_t, d\mathbf{r}_t)$ centrées sur des réalisations hypothétiques $\{\mathbf{x}_t^{(n)}, \mathbf{r}_t^{(n)}\}$ de la variable d'état $(\mathbf{x}_t, \mathbf{r}_t)^T$, également appelées particules. Ainsi, la loi de filtrage $\Pr(d\mathbf{x}_t, d\mathbf{r}_t | \mathbf{y}_{1:t})$ est approchée par l'estimation empirique $P_N(d\mathbf{x}_t, d\mathbf{r}_t | \mathbf{y}_{1:t}) = \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}, \mathbf{r}_t^{(n)}}(d\mathbf{x}_t, d\mathbf{r}_t)$, où $\{\mathbf{x}_t^{(n)}, \mathbf{r}_t^{(n)}\}$ est la $n^{\text{ième}}$ particule et $w_t^{(n)}$ son poids. Si une approximation de $\Pr(d\mathbf{x}_{t-1}, d\mathbf{r}_{t-1} | \mathbf{y}_{1:t-1})$ est connue, alors le procédé d'estimation est divisé en deux étapes :

1. l'étape de diffusion consiste à estimer la densité $p(\mathbf{x}_t, \mathbf{r}_t | \mathbf{y}_{1:t-1})$ en propageant le nuage de particules $\{(\mathbf{x}_{t-1}^{(n)}, \mathbf{r}_{t-1}^{(n)}), w_{t-1}^{(n)}\}_{n=1}^N$ à l'aide d'une distribution d'importance $q(\mathbf{x}_t, \mathbf{r}_t | \mathbf{x}_{0:t-1}^{(n)}, \mathbf{r}_{0:t-1}^{(n)}, \mathbf{y}_t)$;
2. l'étape de mise à jour calcule les nouveaux poids des particules en utilisant les nouvelles observations \mathbf{y}_t : $w_t^{(n)} \propto w_{t-1}^{(n)} \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(n)}, \mathbf{r}_t^{(n)}) p(\mathbf{x}_t^{(n)}, \mathbf{r}_t^{(n)} | \mathbf{x}_{t-1}^{(n)}, \mathbf{r}_{t-1}^{(n)})}{q(\mathbf{x}_t, \mathbf{r}_t | \mathbf{x}_{0:t-1}^{(n)}, \mathbf{r}_{0:t-1}^{(n)}, \mathbf{y}_t)}$. Des techniques de rééchantillonnage sont ensuite utilisées pour éviter les problèmes de dégénérescence du nuage de particules [3].

L'objectif principal de notre approche est de modéliser le vecteur de paramètres \mathbf{r}_t de manière flexible, grâce à un formalisme flou générique, qui peut s'adapter à tous types d'informations, telles que les relations spatiales ou encore les informations de formes. La modélisation floue que nous proposons permet de définir automatiquement la loi *a priori* des paramètres, ainsi que sa transition. Le modèle se suffit donc à lui-même, dans la mesure où ses propriétés *a priori* et de transition se déduisent directement de l'expression des ensembles flous. Cette approche générique permet de simplement modéliser des relations complexes, sans avoir à imposer une modélisation mathématique potentiellement restrictive. Ce travail fait suite à [10], qui a permis, dans la modélisation floue que nous proposons, d'établir une fonction de proposition efficace, qui dépend de la trajectoire des paramètres et des observations courantes.

3 Modélisation des informations floues

La théorie des ensembles flous permet de modéliser des connaissances qualitatives, vagues ou imprécises, souvent exprimées sous forme linguistique [4]. Par exemple, lorsque l'on réalise un sondage auprès d'un ensemble de personnes, en leur demandant quelle est leur taille de manière qualitative, on pourrait aisément regrouper les réponses obtenues selon une granulométrie bien choisie : par exemple *petit*, *moyen* et *grand*. Dans la mesure où ces valeurs linguistiques ne sont que subjectives, donc dépendent des origines, du genre, *etc.* de chacun, il est très probable que les classes obtenues, c'est-à-dire les ensembles flous,

comportent des valeurs communes. Là où une analyse fréquentiste indiquerait avec quelle probabilité un individu tiré aléatoirement dans la population peut mesurer une certaine taille, l'approche floue apporte, généralement, une valeur possibiliste. En traitement des images, et plus particulièrement en suivi d'objets, il est possible de modéliser certaines caractéristiques, ou certains comportements, par une représentation floue. Ainsi, il est par exemple possible de décrire le volume d'un objet par des ensembles flous, avec une granulométrie fixée qui influe sur la précision de la représentation. De même la distance entre les positions d'un objet à deux instants différents peut être caractérisée par les valeurs *proche*, à *moyenne distance*, *éloigné*, dont la sémantique est définie par des ensembles flous sur \mathbb{R}^+ . Enfin, ajoutons, puisque cela fait en partie l'objet de cet article, que des informations de formes floues peuvent être introduites, chaque ensemble flou correspondant à une posture.

3.1 Informations floues

Considérons une famille de K fonctions floues $(f^k)_{k=1}^K$ telle que $\forall k \in \{1, \dots, K\}$, $f^k : \mathcal{Z} \rightarrow [0, 1]$, $\mathcal{Z} \subset \mathbb{R}^{nz}$. On associe à chaque élément $\mathbf{z} \in \mathcal{Z}$ un degré d'appartenance $f^k(\mathbf{z})$ à la classe k . Si l'on se réfère à une sémantique possibiliste, alors $f^k(\mathbf{z})$ correspond au degré de satisfaisabilité de \mathbf{z} à la classe k . Autre exemple, si la classe k définit les éléments impossibles de la $k^{\text{ième}}$ valeur d'un concept flou, alors $f^k(\mathbf{z})$ indique avec quel degré \mathbf{z} traduit cette impossibilité. Dans la terminologie floue, la famille de fonctions $(f^k)_{k=1}^K$ correspond à une variable linguistique dans laquelle chaque fonction f^k est une valeur linguistique possible de la variable. La granularité est définie en fonction de l'application. Ainsi, en considérant une relation spatiale d'orientation floue, les valeurs linguistiques pourraient être *est*, *nord*, *ouest* et *sud*, définies par la famille $(f^k)_{k=1}^4$.

3.2 Fonctions floues

Les fonctions floues $(f^k)_{k=1}^K$ peuvent être fixées, soit par une connaissance *a priori*, donc liée à l'application, soit de manière générique, par apprentissage. C'est dans ce dernier contexte que nous nous plaçons maintenant. Considérons M^k observations que l'on sait appartenir à la classe k , que nous noterons $\{\mathbf{z}_m^k\}_{m=1}^{M^k}$, telles que $\forall m \in \{1, \dots, M^k\}$, $\mathbf{z}_m^k \in \mathcal{Z}$. La $k^{\text{ième}}$ fonction floue \tilde{f}^k peut se définir comme une combinaison floue des éléments \mathbf{z}_m^k :

$$\tilde{f}^k(\mathbf{z}) = \Phi^k(g_{\mathbf{z}_1^k}(\mathbf{z}), \dots, g_{\mathbf{z}_{M^k}^k}(\mathbf{z})) \quad (5)$$

avec $\Phi^k : [0, 1]^{M^k} \rightarrow [0, 1]$ un opérateur de fusion floue, que nous détaillerons dans la section suivante et $g_{\mathbf{z}_m^k} : \mathcal{Z} \rightarrow [0, 1]$ une fonction associée à \mathbf{z}_m^k :

$$g_{\mathbf{z}_m^k}(\mathbf{z}) = \top [P_{\mathbf{z}_m^k}(\mathbf{z}), C_{\mathbf{z}_m^k}(\mathbf{z})] \quad (6)$$

avec \top une t-norme (conjonction floue), $P_{\mathbf{z}_m^k} : \mathcal{Z} \rightarrow [0, 1]$ un patron flou générique connu et fixé et $C_{\mathbf{z}_m^k} : \mathcal{Z} \rightarrow [0, 1]$ des contraintes additionnelles sur la forme de la fonction

floue finale. Celles-ci peuvent être connues *a priori*, ou apprises, comme nous le verrons dans la section applicative de cet article. En l'absence de contraintes, $C_{\mathbf{z}_m^k}$ correspond à l'élément neutre de la t-norme, c'est-à-dire à la valeur 1. Le patron flou $P_{\mathbf{z}_m^k}$ peut par exemple correspondre, dans le cas où $\mathcal{Z} \subset \mathbb{R}^L$, à une boule floue de dimension L :

$$P_{\mathbf{z}_m^k}(\mathbf{z}) = \begin{cases} 1 & \text{si } \|\mathbf{z} - \mathbf{z}_m^k\|_2 \in [0, b] \\ 1 - \frac{\|\mathbf{z} - \mathbf{z}_m^k\|_2 - b}{a} & \text{si } \|\mathbf{z} - \mathbf{z}_m^k\|_2 \in]b; b + a[\\ 0 & \text{sinon} \end{cases}$$

avec $2(a + b)$ la longueur du support et $2b$ celle du noyau. Enfin, nous définissons la fonction floue f^k finale comme une combinaison des fonctions \tilde{f}^k et $\{\tilde{f}^h\} \triangleq \{\tilde{f}^h\}_{h \in \{1, \dots, K\} \setminus \{k\}}$, par un opérateur de fusion $\Psi^k : [0, 1]^K \rightarrow [0, 1]$:

$$f^k(\mathbf{z}) = \Psi^k(\tilde{f}^k(\mathbf{z}), \{\tilde{f}^h(\mathbf{z})\}) \quad (7)$$

Il devient alors possible de définir des interactions entre les ensembles flous en calculant leur degré d'intersection, contraint par un niveau minimum. Soit $R(k, h)$ ce degré entre les ensembles f^k et f^h :

$$R(k, h) = \perp [\Xi^{kh}(f^k, f^h), \epsilon^{kh}] \quad (8)$$

avec \perp une t-conorme (disjonction floue), $\Xi^{kh}(f^k, f^h)$ le degré d'intersection des ensembles f^k et f^h , et ϵ^{kh} le seuil minimal d'intersection des deux ensembles, connu et fixé. Il existe de nombreux opérateurs calculant le degré d'intersection de deux ensembles flous, voir par exemple [1] pour un descriptif des opérateurs existants.

3.3 Stratégies de fusion

Comme nous venons de le voir, les ensembles flous $(f^k)_{k=1}^K$ sont obtenus en combinant différentes stratégies de fusion via les opérateurs $(\Phi^k)_{k=1}^K$, $(\Psi^k)_{k=1}^K$. Le premier est appelé opérateur de fusion intra-classes, car il se réfère à la fusion d'observations issues d'une même classe, tandis que le second fusionne des données de différentes classes, et correspond donc à un opérateur de fusion inter-classes. Il est ainsi possible de décliner une liste non exhaustive des différentes stratégies de fusion envisageables :

Stratégies intra-classes : Φ

- Fusion disjonctive en utilisant des t-conormes : chaque observation représente une vision partielle de l'ensemble des possibles, et donc leur réunion couvre du mieux possible cet ensemble. $f^k(\mathbf{z})$ représente alors dans quelle mesure \mathbf{z} ressemble à au moins une des observations.
- Fusion conjonctive en utilisant des t-normes, intéressante dans au moins deux situations. Lorsque l'on a des observations multiples d'un même phénomène, chacune entâchée d'imprécision, on peut avoir intérêt à dilater les représentations floues correspondantes, pour qu'elles englobent bien la vérité et éviter les conflits entre elles. Dans ce cas la conjonction ne retient que les parties communes, données par toutes les observations, et permet

alors de réduire l'imprécision. Autre situation, lorsque les observations ne donnent pas les possibles mais au contraire des contraintes, dans ces cas on cherche une conjonction des complémentaires des contraintes. Avec une fusion conjonctive, $f^k(\mathbf{z})$ mesure alors l'adéquation entre \mathbf{z} et l'ensemble des observations.

- Fusion de type compromis en utilisant une somme ou une moyenne : cela permet une compensation entre les observations, considérées par exemple comme des évaluations d'une situation, chaque mesure pouvant être imparfaite, on peut les moyennner pour compenser les imperfections de chacune. Avec un opérateur additif, on obtient une approche fréquentiste, et on rejoint les questions d'estimation statistique.

Stratégies inter-classes : Ψ

- Fusion exclusive : si un élément a une appartenance partagée entre plusieurs classes, on considère que c'est une impossibilité sémantique. Cela peut être applicable à la modélisation d'états binaires, pour lesquels il n'existe pas de valeurs intermédiaires.
- Fusion normalisatrice : on normalise l'appartenance de chaque élément à une classe par la somme des appartenances à toutes les classes. Ainsi, $f^k(\mathbf{z})$ peut s'interpréter comme la probabilité d'être dans la classe k sachant l'observation \mathbf{z} .
- Aucune : c'est le comportement par défaut, chaque classe est considérée indépendamment des autres.

Ces stratégies de fusion définissent un cadre générique d'application de notre modèle à l'adaptation de divers problèmes. Ainsi, en fonction de certaines propriétés désirées, ou selon une sémantique des classes particulière, il est possible de combiner ces différentes stratégies afin d'arriver à la représentation souhaitée. Un exemple sera détaillé dans la section 5.

4 Informations floues et filtre particulaire

4.1 Intégration des informations floues dans un contexte probabiliste

Soit \mathcal{Z}^* l'ensemble des possibles

$$\mathcal{Z}^* = \{\mathbf{z} \in \mathcal{Z}; \exists k \in \{1, \dots, K\} \text{ t.q. } f^k(\mathbf{z}) > 0\} \quad (9)$$

et définissons la fonction $\varphi : \{1, \dots, K\} \times \mathcal{Z}^* \rightarrow [0, 1]$ telle qu'elle représente la probabilité d'être dans la classe k sachant \mathbf{z} :

$$\varphi(k, \mathbf{z}) = \frac{f^k(\mathbf{z})}{\sum_{h=1}^K f^h(\mathbf{z})} \quad (10)$$

Alors que $f^k(\mathbf{z})$ représente le degré d'appartenance de \mathbf{z} à la classe k , la modélisation de $\varphi(k, \mathbf{z})$ intègre l'exhaustivité de la représentation par classe, garantissant ainsi le théorème des probabilités totales :

$$\forall \mathbf{z} \in \mathcal{Z}^*, \sum_{k=1}^K \varphi(k, \mathbf{z}) = 1$$

Cette étape permet de passer de la modélisation floue à une sémantique probabiliste pouvant donc être intégrée dans le filtre particulaire. Notons \mathbf{c} la variable aléatoire discrète à valeurs dans $\{1, \dots, K\}$ qui indique l'indice de l'ensemble flou considéré et \mathbf{Z} une variable aléatoire continue à valeurs dans \mathcal{Z}^* . On pose alors les lois suivantes :

$$\Pr(\mathbf{c} = k | \mathbf{Z} = \mathbf{z}) \triangleq \varphi(k, \mathbf{z}) \quad (11)$$

$$\Pr(\mathbf{c} = k) \triangleq \frac{\int_{\mathcal{Z}^*} f^k(\mathbf{z}) d\mathbf{z}}{\sum_{h=1}^K \int_{\mathcal{Z}^*} f^h(\mathbf{z}) d\mathbf{z}} \quad (12)$$

En dérivant les équations 11 et 12, on obtient les densités de probabilités suivantes :

$$p_{\mathbf{Z}}(\mathbf{z}) = \frac{\sum_{k=1}^K f^k(\mathbf{z})}{\sum_{k=1}^K \int_{\mathcal{Z}^*} f^k(\mathbf{z}') d\mathbf{z}'} \quad (13)$$

$$p_{\mathbf{Z}}(\mathbf{z} | \mathbf{c} = k) = \frac{f^k(\mathbf{z})}{\int_{\mathcal{Z}^*} f^k(\mathbf{z}') d\mathbf{z}'} \quad (14)$$

En considérant \mathbf{c}_t un processus stochastique markovien, t étant l'indice du temps, il est possible de définir la transition de \mathbf{c}_{t-1} à \mathbf{c}_t grâce au degré d'intersection entre ensembles flous défini dans l'équation 8 :

$$\Pr(\mathbf{c}_t = k | \mathbf{c}_{t-1} = h) = \frac{R(k, h)}{\sum_{j=1}^K R(k, j)} \quad (15)$$

Ainsi, la transition des variables \mathbf{c}_t se déduit automatiquement de l'expression des fonctions floues, et utilise l'idée qu'il y a plus de chances que \mathbf{c} passe d'une certaine classe h à une classe k lorsque ces deux ensembles flous s'intersectent fortement. À noter que \mathbf{c}_t aura toujours une probabilité maximale de rester dans le même état, puisque dans ce cas, $R(k, k) = \Xi^{kk}(f^k, f^k) \triangleq 1$ si les f^k sont normalisées (c'est-à-dire, $\max_{\mathbf{z}} f^k(\mathbf{z}) = 1$).

4.2 Intégration des informations floues dans un filtre particulaire

Considérons le vecteur de paramètres $\mathbf{r}_t = (\mathbf{r}_t^1, \dots, \mathbf{r}_t^R)$, ses composantes étant indépendantes entre elles. Chaque paramètre \mathbf{r}_t^r est modélisé par une famille de fonctions floues $(f_r^k)_{k=1}^{K_r}$ et est donc défini sur l'ensemble des possibles associé (équation 9). Soit $\mathbf{c}_t = (\mathbf{c}_t^1, \dots, \mathbf{c}_t^R)$ le vecteur désignant à quels ensembles flous les paramètres $(\mathbf{r}_t^1, \dots, \mathbf{r}_t^R)$ sont les plus susceptibles d'appartenir. En considérant ses composantes indépendantes entre elles, on obtient la transition *a priori* de \mathbf{c}_t :

$$\Pr(\mathbf{c}_t = (k^1, \dots, k^R) | \mathbf{c}_{t-1} = (h^1, \dots, h^R)) = \prod_{r=1}^R \Pr(\mathbf{c}_t^r = k^r | \mathbf{c}_{t-1}^r = h^r) \quad (16)$$

avec $\Pr(\mathbf{c}_t^r = k^r | \mathbf{c}_{t-1}^r = h^r)$ la transition décrite dans l'équation 15. Nous modélisons le vecteur de paramètres tel que \mathbf{r}_t soit indépendant conditionnellement à \mathbf{r}_{t-1} sachant \mathbf{c}_t , ce qui nous donne :

$$p(\mathbf{r}_t | \mathbf{c}_t = (k^1, \dots, k^R), \mathbf{r}_{t-1}) = \prod_{r=1}^R p(\mathbf{r}_t^r | \mathbf{c}_t^r = k^r) \quad (17)$$

avec $p(\mathbf{r}_t^r | \mathbf{c}_t^r = k^r)$ la densité décrite dans l'équation 14. La distribution de filtrage au temps t peut être approchée par l'algorithme 1.

Algorithme 1 : Filtre particulaire pour l'intégration d'information spatiale floue

Entrée : $P_N(d\mathbf{x}_{t-1}, d\mathbf{r}_{t-1}, \mathbf{c}_{t-1} | \mathbf{y}_{1:t-1})$

Sortie : Nouvelle loi *a posteriori* $P_N(d\mathbf{x}_t, d\mathbf{r}_t, \mathbf{c}_t | \mathbf{y}_{1:t})$

début

- Propagation des particules : **pour** $n = 1, \dots, N$ **faire**
 générer $\tilde{\mathbf{c}}_t^{(n)} \sim \Pr(\mathbf{c}_t | \mathbf{c}_{t-1}^{(n)})$ selon l'éq. 16.
 générer $\tilde{\mathbf{r}}_t^{(n)} \sim p(\mathbf{r}_t | \tilde{\mathbf{c}}_t^{(n)})$ selon l'éq. 17.
 générer $\tilde{\mathbf{x}}_t^{(n)} \sim q(\mathbf{x}_t | \mathbf{x}_{t-1}^{(n)}, \tilde{\mathbf{r}}_t^{(n)})$.

- Mise à jour des poids : **pour** $n = 1, \dots, N$ **faire**

$$\tilde{w}_t^{(n)} = w_{t-1}^{(n)} \frac{p(\mathbf{y}_t | \tilde{\mathbf{x}}_t^{(n)}, \tilde{\mathbf{r}}_t^{(n)}, \tilde{\mathbf{c}}_t^{(n)}) p(\tilde{\mathbf{x}}_t^{(n)} | \mathbf{x}_{t-1}^{(n)}, \tilde{\mathbf{r}}_t^{(n)})}{q(\tilde{\mathbf{x}}_t^{(n)} | \mathbf{x}_{t-1}^{(n)}, \tilde{\mathbf{r}}_t^{(n)})}$$

- Normalisation des poids : **pour** $n = 1, \dots, N$ **faire**

$$w_t^{(n)} = \frac{\tilde{w}_t^{(n)}}{\sum_{m=1}^N \tilde{w}_t^{(m)}}$$

- Rééchantillonnage si nécessaire.

fin

retourner $\sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}, \mathbf{r}_t^{(n)}, \mathbf{c}_t^{(n)}}(d\mathbf{x}_t, d\mathbf{r}_t, \mathbf{c}_t)$

Dans nos précédents travaux [10], nous avons défini, dans le cadre de relations spatiales floues, une fonction de proposition $q(\mathbf{c}_t | \mathbf{c}_{k:t-1}^{(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_t)$ efficace, qui dépend de la trajectoire de \mathbf{c}_t , de sa transition, de sa loi *a priori* et enfin de l'observation au temps t . Cela rend la simulation des particules plus robuste aux changements rapides de la trajectoire, et permet de lever l'ambiguïté sur des situations complexes grâce à l'information extraite de la trajectoire.

Notons également que la vraisemblance $p(\mathbf{y}_t | \tilde{\mathbf{x}}_t^{(n)}, \tilde{\mathbf{c}}_t^{(n)})$ peut dépendre de l'indice de classes $\tilde{\mathbf{c}}_t^{(n)}$, ce qui permet par exemple de mettre en œuvre des vraisemblances dédiées à la classe considérée. Nous détaillerons plus ce point dans la section 5.

4.3 Approximation

Dans certains cas (stratégie probabiliste, espace \mathcal{Z} de dimension 1, *etc.*), le calcul et/ou la simulation selon les équations 12, 13, 14 et 15 peut se faire de manière analytique (ou directe). Dans les autres cas, il est nécessaire de recourir à une méthode d'approximation. Bien sûr, cette étape nécessite d'être réalisée une seule fois, lors de l'apprentissage des fonctions floues. Nous proposons ici de réaliser l'approximation de la densité *a priori* $p_{\mathbf{Z}}(\mathbf{z})$ via l'algorithme de Metropolis-Hasting indépendant, bien qu'il soit également possible d'utiliser un algorithme d'acceptation-rejet (en modifiant quelque peu la fonction de proposition de simulation). Nous renvoyons à [7] pour une comparaison en termes de convergence de

ces deux algorithmes. En reprenant les notations définies dans la section 3, posons :

$$F(\mathbf{z}) = \sum_{k=1}^K f^k(\mathbf{z}) \quad \text{et} \quad G(\mathbf{z}) = \sum_{k=1}^K \sum_{m=1}^{M^k} P_{\mathbf{z}_m^k}(\mathbf{z})$$

La fonction F correspond à la densité $p_{\mathbf{Z}}(\mathbf{z})$ à une constante près et G à une densité non normalisée fictive, qui serait obtenue avec une stratégie de fusion intra-classe additive, sans stratégie inter-classes ni contrainte $C_{\mathbf{z}_m^k}$. C'est donc une version probabiliste de F , dont le support inclut celui de F , et pour laquelle on sait facilement générer un échantillon. L'approximation peut être réalisée par l'algorithme 2, où $\tau_{\text{début}}$ correspond à la période de "burn-in" et τ_{fin} au nombre d'itérations maximum.

Algorithme 2 : Approximation par Metropolis-Hasting

Entrée : État initial $\mathbf{z}^{(0)}$

Sortie : Approximation de la densité $p_{\mathbf{Z}}(\mathbf{z})$

début

pour $\tau = 0, \dots, \tau_{\text{fin}}$ **faire**

- Simuler un candidat $\mathbf{z}^* \sim G(\mathbf{z})$.

- Le candidat \mathbf{z}^* est accepté avec la probabilité :

$$\alpha = \begin{cases} \min \left\{ 1, \frac{F(\mathbf{z}^*)G(\mathbf{z}^{(\tau)})}{F(\mathbf{z}^{(\tau)})G(\mathbf{z}^*)} \right\} & \text{si } F(\mathbf{z}^{(\tau)})G(\mathbf{z}^*) > 0 \\ 1 & \text{sinon.} \end{cases}$$

- Si accepté, $\mathbf{z}^{(\tau+1)} = \mathbf{z}^*$, sinon $\mathbf{z}^{(\tau+1)} = \mathbf{z}^{(\tau)}$.

fin

retourner $\{\mathbf{z}^{(i)}; Li = l(\tau_{\text{fin}} - \tau_{\text{début}}), l = 1, \dots, L\}$

L'approximation par un échantillon de taille L de la densité *a priori* s'écrit

$$p_{\mathbf{Z}}(\mathbf{z}) = \frac{F(\mathbf{z})}{\int F(\mathbf{z}') d\mathbf{z}'} \approx \frac{1}{L} \sum_{l=1}^L \delta_{\mathbf{z}^{(l)}}(\mathbf{z}) \quad (18)$$

De la même manière, on approche les équations 12 et 14 :

$$p(\mathbf{c} = k) \approx \frac{1}{L} \sum_{l=1}^L \frac{f^k(\mathbf{z}^{(l)})}{\sum_{h=1}^K f^h(\mathbf{z}^{(l)})} \quad (19)$$

$$p_{\mathbf{Z}}(\mathbf{z} | \mathbf{c} = k) \approx \frac{1}{p(\mathbf{c} = k)} \frac{1}{L} \sum_{l=1}^L \frac{f^k(\mathbf{z}) \delta_{\mathbf{z}^{(l)}}(\mathbf{z})}{\sum_{h=1}^K f^h(\mathbf{z})} \quad (20)$$

5 Application au suivi de la bouche dans des séquences d'images de vidéos

Le vecteur d'état $\mathbf{x}_t = (x_t, y_t)$ contient les coordonnées de la bouche à l'instant t . Le vecteur de paramètres $\mathbf{r}_t = (\dot{x}_t, \dot{y}_t, \theta_t, a_t)$ intègre le vecteur de déplacement (\dot{x}_t, \dot{y}_t) , l'orientation de l'objet θ_t et un vecteur de points de contrôle $a_t = (a_t^1, \dots, a_t^P)$. Soit la dynamique de \mathbf{x}_t :

$$\mathbf{x}_t = g_t(\mathbf{x}_{t-1}, \mathbf{r}_t, \mathbf{v}_t) = \begin{cases} x_{t-1} + \Delta_t \dot{x}_t + \mathbf{v}_t^x \\ y_{t-1} + \Delta_t \dot{y}_t + \mathbf{v}_t^y \end{cases}$$

Les paramètres de vélocité, d'orientation et de forme de bouche sont modélisés de manière souple en définissant leurs valeurs possibles par des ensembles flous :

$$\begin{aligned} \mathbf{c}_t^1 &\in \{NO, N, NE, O, Centre, E, SO, S, SE\} \\ \mathbf{c}_t^2 &\in \{N, NE, E, SE, S\} \\ \mathbf{c}_t^3 &\in \{Fermée, Ouverte, Sourire\} \end{aligned}$$

Ainsi, le vecteur de déplacement peut appartenir à neuf ensembles flous, du *Nord-Ouest* au *Sud-Est*, l'orientation de l'objet, du *Nord* au *Sud*, et les points de contrôle aux postures floues *Fermée*, *Ouverte* et *Sourire*.

5.1 Modélisation des fonctions floues

On appelle partition uniforme floue de \mathcal{H} une famille de fonctions floues $(f^k)_{k=1}^K$ qui satisfait les deux propriétés suivantes :

- $\forall x \in \mathcal{H}, \sum_{k=1}^K f^k(x) = 1$
- $\forall (k, h) \in \{1, \dots, K\}^2, \int_{\mathcal{H}} f^k(x) dx = \int_{\mathcal{H}} f^h(x) dx$

Dans notre modèle, cela signifie que les lois $Pr(\mathbf{z} \in \mathbf{Z})$ et $Pr(\mathbf{c} = k)$ sont uniformes et que $f^k(\mathbf{z})$ s'interprète aussi comme la probabilité d'être dans la classe k sachant \mathbf{z} . Nous modélisons la famille de fonctions floues correspondant à la vélocité de l'objet comme une partition uniforme floue de $[-h_x, h_x] \times [-h_y, h_y]$, avec h_x et h_y des paramètres d'horizon fixés, et celle de l'orientation comme une partition uniforme floue de $[-\pi/2, \pi/2]$. En considérant des fonctions floues trapézoïdales, il est ainsi possible de simuler et calculer de manière analytique les équations 12, 13, 14 et 15. Afin de faire le parallèle avec les stratégies de fusion évoquées dans la section 3.3, il aurait tout à fait été possible de définir ces classes par apprentissage, en utilisant une stratégie intra-classe disjonctive et une stratégie inter-classes normalisatrice.

La famille des fonctions de posture de la bouche est obtenue automatiquement par apprentissage. Soit $BD = \{(a^{(m)}, c^{(m)}), m = 1, \dots, M\}$ une base de données d'éléments $a^{(m)} \in \mathbb{R}^{2P}$ de P points de contrôle 2D annotés d'une étiquette $c^{(m)} \in \{1, \dots, K\}$. L'algorithme d'apprentissage des fonctions floues se résume par :

1. Création de la base de données :
 - Collecte des M échantillons $\{(a^{(m)}, c^{(m)}), m = 1, \dots, M\}$.
 - Rotation des échantillons et mise à l'échelle (facultative), en utilisant une analyse en composantes principales.
 - Répartition des échantillons par classe.
2. Création des fonctions floues associées aux échantillons $\{g_{a^{(m)}}\}_{m=1}^M$ (équation 6).
3. Création des fonctions floues de classes $(f^k)_{k=1}^K$ (équation 7).
4. Si besoin, faire l'approximation de la densité de probabilité 13 via la méthode d'approximation décrite dans la section 4.3.

Dans l'application proposée, on utilise une fusion intra-classe disjonctive, avec des t-conormes ($\Phi = \max$), sans fusion inter-classes (donc $f^k = \tilde{f}^k$). Ce choix mène à des classes de sémantique possibiliste, ce qui semble bien adapté, entre autres, aux applications d'apprentissage : la base de données représentant un échantillon des réalisations possibles, chacun de ses éléments est donc possible, réalisable. On souhaiterait intégrer aux fonctions floues $g_{a^{(m)}}$ des contraintes de forme, afin d'être en mesure de définir une classe floue, mais qui reste cohérente avec les formes considérées. Pour ce faire, plusieurs stratégies peuvent être mises en œuvre, en définissant les contraintes de forme $C_{a^{(m)}}$. D'une manière générique, cette fonction floue peut s'écrire :

$$C_{a^{(m)}}(\mathbf{z}) = \Lambda(d(\mathbf{z}, a^{(m)}))$$

avec $\Lambda : \mathbb{R}^+ \rightarrow [0, 1]$ une fonction floue décroissante et $d : \mathbb{R}^{2P} \times \mathbb{R}^{2P} \rightarrow \mathbb{R}^+$ une distance entre deux formes. Afin de garantir une forme globale quasi identique à celle de l'échantillon $a^{(m)}$, tout en autorisant des changements d'échelles et des changements mineurs locaux, nous définissons la distance de forme par :

$$d(x, y) = \frac{1}{P} \sum_{p=1}^P \left| \widehat{(x^{p-1}, x^p, x^{p+1})} - \widehat{(y^{p-1}, y^p, y^{p+1})} \right|$$

avec $\widehat{(a, b, c)}$ l'angle entre les vecteurs \vec{ab} et \vec{bc} .

D'autres propriétés pourraient être obtenues en utilisant d'autres types de distance. Par exemple, une distance entre histogrammes de directions impliquerait une invariance par rotation, translation et changement d'échelle.

En utilisant un patron flou $P_{a^{(m)}}$ assez permissif, par exemple correspondant à une boule floue, et des contraintes $C_{a^{(m)}}$ fortes, cela permet d'obtenir une classe étalée, tout en restreignant les formes. Des exemples de simulation de bouche sont illustrés sur la figure 1, en utilisant différents paramètres de patron et de contraintes de forme, et menant ainsi à des degrés d'intersection entre les classes plus ou moins forts (tableau 1), qui sont directement liés, comme nous l'avons vu, à la transition entre classes (équation 15). Soit le degré d'intersection des classes k et h :

$$\begin{aligned} \Xi(f^k, f^h) &= \frac{\int_{\mathcal{Z}} \Upsilon(f^k(\mathbf{z}), f^h(\mathbf{z})) d\mathbf{z}}{\min \left[\int_{\mathcal{Z}} \mathbb{1}^k(\mathbf{z}) d\mathbf{z}, \int_{\mathcal{Z}} \mathbb{1}^h(\mathbf{z}) d\mathbf{z} \right]} \\ &\approx \frac{\sum_{l=1}^L \Upsilon(f^k(\mathbf{z}^{(l)}), f^h(\mathbf{z}^{(l)}))}{\min \left[\sum_{l=1}^L \mathbb{1}^k(\mathbf{z}^{(l)}), \sum_{l=1}^L \mathbb{1}^h(\mathbf{z}^{(l)}) \right]} \end{aligned}$$

avec $\Upsilon(a, b)$ un opérateur flou tel que $\Upsilon(a, b) = 1$ si $a \times b \neq 0$ et 0 sinon, et $\mathbb{1}^k(x) = \Upsilon(f^k(x), 1)$ la fonction indicatrice de la classe k . Le degré d'intersection défini ici est souple, dans le sens où il accentue volontairement le degré d'intersection entre deux classes, afin de favoriser les transitions définies dans l'équation 15. Les résultats du tableau 1 montrent que les degrés d'intersection obtenus avec un patron large avec contraintes de forme sont plus



FIG. 1 – Simulations selon $p_{\mathbf{z}}(\mathbf{z}|\mathbf{c}_t^3 = 3)$ avec un patron (a) de support restreint, (b) de support large et (c) de support large avec contraintes de forme. Le rectangle rouge correspond au support d'un point de contrôle ($P = 18$).

élevés qu'avec un patron simplement large. Bien que cela puisse paraître contre-intuitif, cela s'explique avec le dénominateur intervenant dans le calcul du degré d'intersection. Cela signifie donc que les fonctions floues, bien qu'étant plus restreintes, restent larges aux endroits où les formes sont cohérentes vis-à-vis des contraintes imposées, et plus particulièrement dans les zones d'appartenance partagées entre plusieurs classes.

$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	(a)	$\begin{bmatrix} 1 & 0,12 & 0,01 \\ 0,12 & 1 & 0,07 \\ 0,01 & 0,07 & 1 \end{bmatrix}$	(b)	$\begin{bmatrix} 1 & 0,18 & 0,02 \\ 0,18 & 1 & 0,13 \\ 0,02 & 0,13 & 1 \end{bmatrix}$	(c)
---	-----	---	-----	---	-----

TAB. 1 – Matrices de degrés d'intersection avec un patron (a) de support restreint, (b) de support large et (c) de support large avec contraintes de forme. Les lignes et les colonnes représentent les indices des ensembles flous.

5.2 Fonctions de vraisemblance

Afin que le suivi soit robuste, nous proposons d'utiliser une vraisemblance formée de deux composantes, l'une utilisant les régions, et l'autre les contours :

$$p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{r}_t, \mathbf{c}_t^3 = k) = p(\mathbf{y}_t^R|\mathbf{x}_t, \mathbf{r}_t, \mathbf{c}_t^3 = k) p(\mathbf{y}_t^C|\mathbf{x}_t, \mathbf{r}_t)$$

avec $p(\mathbf{y}_t^C|\mathbf{x}_t, \mathbf{r}_t)$ fonction du gradient de l'image lissée :

$$p(\mathbf{y}_t^C|\mathbf{x}_t, \mathbf{r}_t) \propto \exp\left(\lambda_1 * \frac{1}{P} \sum_{p=1}^P \nabla [I_t(a_t^p) \star \mathcal{N}(a_t^p, \sigma)]\right)$$

avec $\lambda_1 \in \mathbb{R}^+$ une valeur constante fixée, ∇ l'opérateur de gradient, \star l'opérateur de convolution, $I_t(x)$ le niveau de gris au point x de l'image I_t au temps t et $\mathcal{N}(x, \sigma)$ une loi normale de moyenne x et de variance fixée σ . En pratique, nous interpolons par une *b-spline* les points de contour, et calculons la réponse du filtre sur l'ensemble des points interpolés, afin que la réponse du filtre soit plus robuste.

La vraisemblance utilisant les régions prend en compte les indices des classes c_t . Elle est liée d'une part à une distance entre histogrammes d'un modèle de bouche $h_k^\bullet = \{h_k^\bullet(u)\}_{u=1}^U$ de posture k , et un histogramme candidat $h(\hat{a}_t) = \{h(\hat{a}_t; u)\}_{u=1}^U$ [8], avec u l'indice d'une case d'un histogramme de longueur U , et d'autre part à une distance entre l'histogramme candidat et l'histogramme $h_0^\bullet = \{h_0^\bullet(u)\}_{u=1}^U$ de l'arrière plan de l'image :

$$p(\mathbf{y}_t^R|a_t, \mathbf{c}_t^3 = k) \propto \exp(-\lambda_2(d^2[h_k^\bullet, h(\hat{a}_t)] - d^2[h_0^\bullet, h(\hat{a}_t)]))$$

avec $\lambda_2 \in \mathbb{R}^+$ une valeur constante fixée et \hat{a}_t l'objet associé au contour a_t . La distance utilise le coefficient de

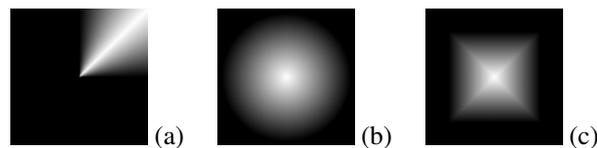


FIG. 2 – Relations spatiales floues (a) orientation *nord-est* et *quart centre* avec (b) une boule floue et (c) un carré flou.

Bhattacharyya et est définie par :

$$d[h_k^\bullet, h(\hat{a}_t)] = \left[1 - \sum_{u=1}^U \sqrt{h_k^\bullet(u) h(\hat{a}_t; u)}\right]^{1/2}$$

Les histogrammes de références $\{h_k^\bullet\}_{k=0}^3$ sont créés pendant la phase d'apprentissage, comme la moyenne des histogrammes obtenus en utilisant plusieurs bouches issues de la base de données.

5.3 Expérimentations

L'apprentissage et les tests sont issus d'une base de données disponible librement sur Internet¹. Les fonctions floues ont été obtenues en considérant trois classes de sept échantillons chacune. Les tests ont été réalisés sur les 180 premières images. Nous avons utilisé un patron large avec des contraintes de formes (figure 1(c)) en utilisant un nombre de points de contrôle $P = 18$, générant les probabilités de transition décrites sur le tableau 1 (c), sans utiliser de seuil minimal d'intersection entre classes (équation 8). Les partitions floues des concepts d'orientation et de vélocité sont également de type triangulaire (de représentation spatiale illustrée sur la figure 2), avec une valeur d'intersection minimale $\epsilon^{kh} = 0,05$, afin d'autoriser des changements brusques de valeurs des paramètres.

Pour la vraisemblance, nous avons choisi un espace colorimétrique décomposé en teinte T , saturation S et valeur V . L'histogramme est de longueur $U = U_T \times U_S \times U_V$, avec $U_T = U_S = U_V = 10$ le nombre de cases par composantes. Les constantes multiplicatives λ_1 et λ_2 ont été fixées à 20.

	Aire des différences (en %)			
	100	500	2000	5000
Modèle proposé	23,8	20,1	19,4	19,3

TAB. 2 – Erreur moyenne obtenue avec $N = 100, 500, 2000$ et 5000 particules.

Le tableau 2 présente les erreurs moyennes de recouvrement obtenues en utilisant un nombre croissant de particules tandis que la figure 4 illustre les erreurs observées sur la séquence entière avec $N = 5000$ particules. Nous pouvons également constater les postures estimées sur la figure 5. La comparaison avec la vérité terrain montre que notre approche obtient, en majorité, les bonnes postures (1

¹http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/data/talking_face/talking_face.html



FIG. 3 – Estimation obtenue aux instants (a) $t = 11$, (b) $t = 66$, (c) $t = 140$ et (d) $t = 146$.

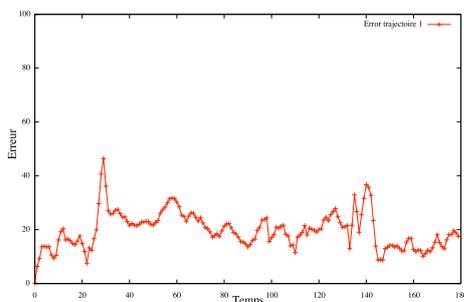


FIG. 4 – Erreur obtenue avec $N = 5000$ particules sur 180 images. L'erreur est calculée comme l'aire des différences entre le résultat obtenu et la vérité terrain, en pourcentage.

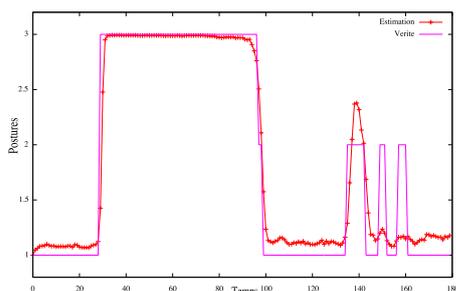


FIG. 5 – Postures obtenues avec $N = 5000$ particules et 180 images.

pour bouche fermée, 2 pour ouverte et 3 pour sourire). Bien qu'on puisse observer un léger accroissement des valeurs, les postures obtenues aux images ~ 150 et ~ 160 sont mises en échec pour une raison simple : cette période correspond à des passages où l'individu entrouvre très légèrement la bouche. La vérité terrain n'intègre pas une information de degré d'ouverture de la bouche, ce qui explique ce résultat. Enfin, l'espérance de Monte-Carlo obtenue dans notre séquence de tests est illustrée à divers temps sur la figure 3.

6 Conclusion

Nous avons présenté dans cet article une méthode d'intégration d'informations floues dans le filtre particulaire. L'approche proposée est générique, au sens où elle permet d'utiliser divers types d'informations. Dans notre application, nous avons utilisé des informations structurales spatiales, connues et fixées, ainsi que des informations floues de formes, apprises. Nous avons ainsi proposé une méthode

simple d'apprentissage de formes floues, adaptable à n'importe quel type de forme, tant les descripteurs de formes sont basiques : il s'agit uniquement des points de contrôle. Toutefois, afin d'intégrer à la fonction floue des contraintes de forme cohérentes, nous contraignons sa forme en faisant appel à une distance de forme, qui peut également être définie de manière arbitraire. Enfin, le modèle flou est intégré au filtre particulaire sous la forme d'estimation des paramètres, et peut être couplé à des techniques de simulations par échantillonnage d'importance plus sophistiquées, comme nous l'avons proposé dans des précédents travaux [10]. Une extension intéressante consisterait à adapter notre modèle au suivi multi-objets, en proposant par exemple l'intégration de relations spatiales entre objets.

Références

- [1] I. Bloch. *Fuzzy spatial relationships for image processing and interpretation: a review*. Image and Vision Computing. 23(2) : 89-110, 2005.
- [2] W.Y. Chang, C.S. Chen, Y.D. Jian. *Visual Tracking in High-Dimensional State Space by Appearance-Guided Particle Filtering*. IEEE Trans. on Image Processing. 17 (7) : 14, 2008.
- [3] A. Doucet, N. De Freitas, N. Gordon. *Sequential Monte Carlo methods in practice*. Springer-Verlag, 2001.
- [4] D. Dubois and H. Prade. *Fuzzy Sets and Systems : Theory and Applications* Academic Press, 1980.
- [5] M. Isard, A. Blake. *CONDENSATION – conditional density propagation for visual tracking*. Int. J. Computer Vision. 29 (1) : 5-28, 1998.
- [6] R.E. Kalman *A New Approach to Linear Filtering and Prediction Problems*. Trans. of the ASME - Journal of Basic Engineering. 82 : 35-45, 1960.
- [7] J.S. Liu. *Metropolized Independent Sampling with Comparisons to Rejection Sampling and Importance Sampling*. Statistics and Computing. 6 (2) : 113-119, June 1996.
- [8] P. Pérez, C. Hue, J. Vermaak, M. Gangnet. *Color-Based Probabilistic Tracking*. Proc. of the 7th European Conference on Computer Vision-Part I. 661-675, 2002.
- [9] Y. Rathi, N. Vaswani, A. Tannenbaum. *A Generic Framework for Tracking Using Particle Filter With Dynamic Shape Prior*. IEEE Trans. on Image Processing. 16 (5) : 1370-1382, 2007.
- [10] N. Widynski, S. Dubuisson et I. Bloch, *Intégration de relations spatiales floues dans un filtre particulaire pour le suivi d'objets*, GRETSI. Dijon, France, Sept. 2009.